PATENT ABSTRACTS OF JAPAN

(11)Publication number : 10-083310

(43)Date of publication of application : 31.03.1998

(51)Int.Cl. G06F 9/445
G06F 9/06
G06F 12/14

(21)Application number : 09-151768

(22)Date of filing : 10.06.1997

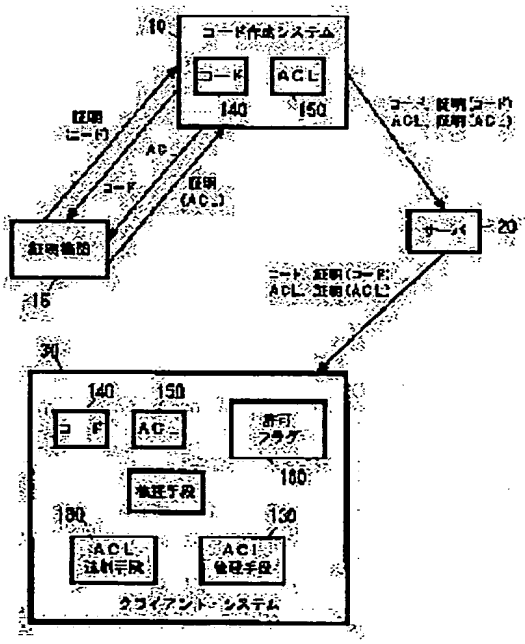
(71)Applicant : INTERNATL BUSINESS MACH CORP <IBM>

(72)Inventor : DAN ASIT
RAMASWAMI RAJIV
SITARAM DINKAR

(30)Priority

Priority number : 96 661517 Priority date : 11.06.1996 Priority country : US

(54) PROGRAM CODE DISTRIBUTING METHOD AND ITS SYSTEM



(57)Abstract:

PROBLEM TO BE SOLVED: To provide an authentification system for allowing a relied third party to confirm the author of a program and to sign a certification for guaranteeing the perfection of the program.

SOLUTION: A program code 140 is capsuled together with a guarantee and access control list(ACL) 150. ACL 150 describes an allowable condition and a resource required by the code 140. A forcing mechanism assigns the allowanble condition of a system and a resource according to ACL 150. For example, a code preparing system 10 communicates with a certifying organization 15 being the relied third party. The organization 15 issues the cerificate of the code 140 and the certificate of ACL 150 of the code 140. Once the certificate is issued, nobody can change the code 140 and ACL 150 without invalidating the certificate. The code 140, its ACL 150 and their certificates are stored in a server.

CLAIMS

[Claim(s)]

[Claim 1] The distribution approach of the program code which is the distribution approach of a program code and is characterized by to describe a resource and permissive conditions required for the harmless actuation in which the above-mentioned program code was inspected possible [reading] by the computer in certification of the above-mentioned third person by whom reliance is done including the step which offers the certification of the third person trusted to a reception system.

[Claim 2] The distribution approach of the program code which contains the step which encapsulates the above-mentioned certification with a program code in claim 1.

[Claim 3] The distribution approach of the program code which contains the resource of the above-mentioned reception system, and the step which assigns permissive conditions in claim 1 so that the step in which the above-mentioned reception system reads the above-mentioned certification, and the permissive conditions by which it was specified during the above-mentioned certification may not be exceeded.

[Claim 4] The distribution approach of the program code containing the step which denies or affirms access and permissive conditions of the above-mentioned program code to the resource of the above-mentioned reception system in claim 2 according to the option which the user chose in relation to the above-mentioned certification.

[Claim 5] The distribution approach of the program code characterized by offering the above-mentioned resource described possible [reading] by computer, and permissive conditions in a code format in claim 1 in the above-mentioned reception system.

[Claim 6] The distribution approach of a program code that the enciphered inspection data are contained in the above-mentioned certification in claim 1, and the above-mentioned reception system is characterized by decoding the above-mentioned inspection data and inspecting the integrity of description of the above-mentioned resource and permissive conditions.

[Claim 7] The distribution approach of the program code characterized by describing both the amount of each resource used by the above-mentioned program code into required description of a resource, and the maximum-permissible specific consumption of the resource in claim 3.

[Claim 8] The distribution approach of the program code characterized by describing the specific function of the above-mentioned reception system accessed by the above-mentioned program code into required description of permissive conditions in claim 3.

[Claim 9] The distribution approach of the program code characterized by describing the functionality of the above-mentioned program code in certification of the above-mentioned third person according to inspection by the above-mentioned third person in claim 1.

[Claim 10] The distribution approach of the program code characterized by the above-mentioned program code being the applet downloaded as a program object from a server in claim 1.

[Claim 11] The distribution approach of the program code characterized by assigning a resource

which is different in claim 1 to the user from whom the above-mentioned program code differed in the above-mentioned reception system, and permissive conditions.

[Claim 12] The distribution approach of a program code that the group of the resource which a user different the account of a top is allowed in claim 11, and permissive conditions is characterized by being determined at the time of install of the above-mentioned program code.

[Claim 13] The distribution approach of a program code that the group of the resource which a user different the account of a top is allowed in claim 11, and permissive conditions is characterized by being determined at the time of activation of the above-mentioned program code.

[Claim 14] In the certification of the program code by the third person who is the distribution approach of a program code and is trusted What described a resource and permissive conditions required for the harmless actuation in which this program code was inspected possible [reading] by computer is put in. The step which encapsulates the certification with the above-mentioned program code, and is offered to a reception system in a code format, The step in which the above-mentioned reception system reads the certification of the above-mentioned code format, The account of a top The step which determines the integrity of the read certification by the above-mentioned reception system, The step which assigns the resource and permissive conditions of the above-mentioned reception system according to the option which the user chose so that the permissive conditions specified in the above-mentioned certification might not be exceeded, after the above-mentioned integrity was inspected, the step which performs the above-mentioned program code according to the above-mentioned assignment -- containing -- the above -- in required description of a resource Both the amounts and maximum-permissible specific consumption of each resource used by the above-mentioned program code are described. In description of the above-mentioned permissive conditions The distribution approach of the program code characterized by describing the specific function of the above-mentioned reception system accessed by the above-mentioned program code.

[Claim 15] The import equipment which imports a program and data to computer system, The operating system which controls actuation of the above-mentioned computer system, The access logic which extracts what described the resource required for the harmless actuation in which the code was inspected possible [reading] by computer from the above-mentioned data, and relates it with a given program, The integrity inspection logic which generates the inspection data in which integrity is shown although it was contained in this access logic and the above-mentioned computer described possible [reading], Connect with the above-mentioned operating system, answer the above-mentioned inspection data, and consumption and specific consumption are pursued and assigned in the above-mentioned reception system about each of many resources. Computer system possessing the compulsive logic it is made not to exceed the assignment specified in the above-mentioned description, and the processor which performs a program code according to the above-mentioned assignment.

[Claim 16] The DS memorized by random access memory is connected to the above-mentioned compulsion logic in claim 15. In this DS The 1st field which pursues the consumed actual resource,

and the 2nd field which pursues the specific consumption of a resource, Computer system characterized by including the 3rd field which memorizes the limit of the resource consumption pulled out from the above-mentioned description, and the 4th field which memorizes the limit of the resource specific consumption pulled out from the above-mentioned description.

[Claim 17] Computer system characterized by including a means to un-encapsulate the above-mentioned description in claim 15 from the package which contains a program code in the above-mentioned access logic, and a means to decode this description.

[Claim 18] Computer system characterized by including a means to deny or affirm code access to the resource of computer system, and permissive conditions in claim 15 according to the option which the user chose in relation to the above-mentioned description into the above-mentioned compulsion logic.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[Field of the Invention] Generally this invention relates to sending software using a distribution system like a network.

[0002]

[Description of the Prior Art] The distribution business of software is the latest big industry. Software is distributed through a diskette and CD-ROM, and came to be gradually distributed through the network today. Furthermore, only through the Internet, from a remote site, a client can download an applet from a server and, recently, it it not only downloads a code, but can be performed. This is a paradigm proposed in Java (Java) and programming language, or other language like Telescript.

[0003] The important problem accompanying the code obtained from others is a problem of security. For example, in a UNIX operating system, a code is performed in the client shell possessing all the privileges of a client. Accessing all the files of a client, sending e-mail, and trying trespass further are also included in such a privilege. Java performs an applet in the environment restricted very much, in order to solve the problem of such an applet. Consequently, the usefulness and functionality of an applet of Java are restricted. On the other hand, the application of Java does not get used to any authentications (authentication) depending on the security offered by the operating system. Therefore, these applications raise the problem of the same security as other codes.

[0004] One format of authentication is proposed by "Trusted Distribution ofSoftware Over the Internet" (A. D. Rubin, Internet Society Symposium - Network and Distributed Security, 1995). There, the integrity of a program is guaranteed, while the third person trusted signs certification and checks the author of a program.

[0005]

[Problem(s) to be Solved by the Invention] Although a client can verify authentication of a code by this approach, this approach does not specify the permissive conditions relevant to a code flexibly, and does not offer how a client forces these permissive conditions automatically, either.

[0006]

[Means for Solving the Problem] The program code is encapsulated, although this invention guarantees the integrity of a program while the third person trusted signs certification and checks the author of a program (that is, related with certification and an access control list). The access control list has described the permissive conditions and the resource which are demanded in code. Furthermore, this invention offers the compulsive mechanism which assigns permissive conditions and a resource to a system according to an access control list.

[0007] A code creation system communicates with the certification engine which is the third person trusted in an example. A certification engine publishes certification of a code, and certification of the access control list about the code. Once certification is published, any persons cannot change the code or access control list, without making the certification into an invalid. A code and its access control list are memorized by the server with those certification. The client which downloads a code or an access control list can inspect the integrity of a code thru/or an access control list, and a system can carry out an access control list so that permissive conditions and a resource may not be exceeded.

[0008]

[Embodiment of the Invention] Drawing 1 is the block diagram of the code sending according to this invention, and a certification system. One or more code creation system 10, certification engine 15, one or more servers 20, and one or more client systems 30 are shown there. A client system is connected to a server through a usual wide area network (WAN) or a usual local area network (LAN). A code creation system contains the program code 140 to prove with a certification engine, and the access control list (ACL) 150 about the code. A certification engine offers the public key K which is used widely and known by the client system. Furthermore, the certification engine holds the private key P which only it knows. In order to offer the certification of a code, a certification engine creates certification including the code hash (hash) of a code name and a code, and signs it using a private key. Now, the certification cannot be changed, without making a signature of a certification engine into an invalid. Similarly, a certification engine also signs the certification for access control lists relevant to a code (supposing it wishes, you may make it sign only one certification for the both sides of a code and its access control list). In the example of drawing 1 , a client system receives certification, an access control list, and a code through WAN or LAN. However, please change to the approach and notice a client system about the point that the proved code is receivable, through a dismountable storage. Such a storage is read by local import equipment like a floppy drive or an optical disk drive.

[0009] The client system shown in drawing 2 includes the inspection means 110, the access control list (ACL) management tool 130, the activation means 170, the access control list (ACL) compulsion means 180, and the client interface 190. Furthermore, a client system contains the usual CPU and

the related control logic 162, the communication link controller / network interface 194, the CD-ROM / floppy disk subsystem 196, and other resources (for example, a network access module, a display subsystem, and various kinds of special purpose adapters). Although it will probably be clear for this contractor that a client system contains many other usual components, they are not explained to a detail here.

[0010] As for the access control list management tool 130 and the access control list compulsion means 180, it is desirable to carry out as a program code. The actuation is explained to a detail later. An activation means is the usual part of the operating system (not shown) of a client, and it is used in order to perform the code imported on the client system. The client interface 190 can be carried out as a front of a screen graphical user interface, and enables communication with the access control list management tool 130 and a client. (For example, by it, a client can order the access control list management tool 130 whether to allow program access to the specified resource, or not to allow, or can manage program access.) As for the inspection means 110, it is desirable to carry out as a program code. It inspects authentication of the code imported with the access control list including a decode module. Furthermore, an inspection means contains a hashing (hashing) module. This module checks the integrity of the code and access control list which were imported.

[0011] "A code / access control list, and certification" If 100 downloads, the sign of the certification engine which the inspection means 110 has after certification will inspect first whether it is the right (a certification engine's known public key is used). Next, it inspects whether an inspection means calculates the code hash of a code/access control list, and its it corresponds with the value under certification. When a sign is not right, the code and access control list are refused (refusal step 120). (namely, when a hash is not in agreement) If inspection is O.K., the access control list management tool 130 will be called. An access control list management tool displays an access control list (it explains later) on a client through the client interface 190, and a client confirms [to which each item in an access control list is permitted / or or] whether authorization is carried out. The access control list management tool 130 memorizes a code 140, and memorizes an access control list 150 with the authorization flag 160 as it was ordered from the client through the client interface.

[0012] There are a file system, a specific file and a logic resource like a system call, and a physical resource like assignment of desk space, disk access, and main memory and access to various kinds of system controllers and adapters as resource which can control access by the access control list compulsion means 180. About access to a logic resource (function of a client system), the authorization flag 160 indicates [to which each item is permitted / or or] whether authorization is carried out. About access to a physical resource, in order to show the maximum permissible dose or maximum-permissible specific consumption, an authorization flag can be used.

[0013] Furthermore, as for drawing 2 , the part from which the client system differed shows how it operates mutually. In a multiple user client system, each user can have each authorization flag group. Furthermore, an environment variable can be put in into an access control list. By changing an environment variable during activation, each user can customize an access privilege. An access

control list and an authorization flag are memorized to a security field, and read-out and updating of this field are a privilege which the access control list compulsion means 180 is allowed.

[0014] An access control list management tool may be called by the client at the time of arbitration, in order to display or change the authorization conditions of an access control list. A code is performed by the activation means 170. Before allowing access to a resource, an activation means calls an access control list compulsion means, and checks the effectiveness of access. This is attained by the approach the activation means 170 inserts into a code the trap to the test routine which calls an access control list compulsion means. Actuation of an access control list compulsion means is explained to a detail later.

[0015] In a system, if needed, separately, it can combine and a code and its access control list can be downloaded. For example, as a plan of a code creation system, although an access control list is offered to all clients for nothing, the code itself can be made into the charge.

[0016] Drawing 3 shows an access control list. An access control list consists of two parts. That is, they are the physical resource table 200 containing the physical resource demanded in code, and the logic resource table 250 containing the authorization conditions demanded in code, and a logic resource.

[0017] The line about each resource is in the physical resource table 200, and the physical resource name 205, a resource attribute 210, the maximum-permissible specific consumption 215, and the maximum permissible dose 220 are in this line. A resource attribute 210 is used when two or more attributes are in a physical unit or a resource disk. For example, it is a case as there are a tooth space and the number of I/O about storage. The maximum-permissible specific consumption 215 and the maximum permissible dose 220 are a resource, the maximum-permissible specific consumption of an attribute, and maximum-permissible consumption.

[0018] The logic resource table 250 contains one line about each call to the external routine (called a logic resource) demanded in code. There are the logic resource name 255 and a parameter list 260 in each line. A parameter list 260 points to the list of parameter entries 265. Each parameter entry 265 specifies the set of the effective parameter range. That is, the set is a set of the effective parameter value as a combination. In the parameter entry 265, the nextPE field 280 indicating the following parameter entry is also included. The parameter range of each parameter includes the two fields. That is, it is the parameter value 275 which specifies the scope of the parameter as the parameter type 270. About a string parameter, the parameter type 270 is STR, and parameter value 275 is a list of regular expressions which specify the effective format about the string. About an integer parameter, the parameter type 270 is INT, and parameter value 275 is a list of integer range. The identifier of an environment variable can be put in into parameter value 275, and permuting the value of the environment variable in that case at the time of activation is assumed.

[0019] The authorization conditions and resource which were specified about the code in the access control list are offered, and the access control list compulsion means in a client ensures that the addition of an authorization condition / resource is not allowed.

[0020] Compulsion of an access control list may be static, or may be dynamic. In static compulsion,

before a code is performed, compulsion is performed extensively, and compulsion while the code is performed is not needed. In dynamic compulsion, while the code is performed, coercion must be exerted. The certification engine itself inspects an access control list, and if it is guaranteed that the excess in a code does not arise, the compulsive function of an access control list will become unnecessary by the client system.

[0021] Drawing 4 shows the DS used by the access control list compulsion means 180. One line is shown in the execution-time physics resource table 300 about each resource. A resource name 300, a resource attribute 310, the maximum-permissible specific consumption 315, and the maximum permissible dose 320 are the copies of the field where the physical resource table 200 corresponds. It is actually used in order to pursue the actual specific consumption of specific consumption 325 and the code [actually / field / of the amount used 330 / each] at the time of activation respectively, and actual consumption. The line about each logic resource needed is shown in the execution-time logic resource table 350. An execution-time logic resource table is the copy of a logic resource table, and the authorization flag is added to it. An authorization flag shows [to which the combination is permitted by the access control list management tool / or or] whether authorization is carried out about each effective combination of a parameter. The logic resource name 355 is the copy of the field where the logic resource table 250 corresponds, and a parameter list 360 points to the list of execution-time parameter entries 365. The parameter type 370 and parameter value 375 of an execution-time parameter entry are the copy of the field where it corresponds in a logic resource table. Yes which shows [to which this execution-time parameter entry 365 is permitted / or or] whether authorization is carried out, the nextPE field 380 points to the following execution-time parameter entry 365, and the authorization flag 385 is set as the no. Furthermore, an access control list compulsion means pursues 395 at the code initiation time.

[0022] Drawing 5 shows how the authorization conditions of a logic resource are forced by the access control list compulsion means. This pass is always called, when a code performs the call to an external function. At step 410, the access control list compulsion means 180 determines the location of the number of parameters, the value of a parameter, and the identifier of the function currently called. The actual approach changes with processors. For example, in Java, these are placed on the operand stack. At step 415, an access control list compulsion means determines the location of the line of the function currently called in the execution-time logic access table 350, and determines the location of the first execution-time parameter entry 365 where the authorization flag is set as yes. When a function name is not found or such an execution-time parameter entry 365 does not exist, an access control list compulsion means progresses to step 470, and it is shown that a call is not allowed and it ends. At step 420, an access control list compulsion means determines the location of the value of the first parameter, and makes it a current parameter. At step 425, an access control list compulsion means confirms whether the value of a current parameter is allowed by the execution-time parameter entry 365. If the value is allowed, it will be confirmed at step 430 whether an access control list compulsion means has a parameter more. If there is furthermore a parameter, at step 435, an access control list compulsion means will set a current parameter as the

following parameter, and will return to step 425. At step 430, if there is no parameter beyond it, an access control list compulsion means progresses to step 450, will display the purport allowed a call and will be completed.

[0023] If the test of admissibility fails in step 425, an access control list compulsion means will progress to step 445 at it. There, an access control list compulsion means checks the execution-time logic resource table 350, and other execution-time parameter entries by which the authorization flag 385 is set to yes are looked for. If there is such no execution-time parameter entry, an access control list compulsion means progresses to step 470, will display the purport which is not allowed a call and will be completed. If there is such an execution-time parameter entry 365, an access control list compulsion means will progress to step 420.

[0024] Drawing 6 shows how physical requirements are forced by the access control list compulsion means. Before an access control list compulsion means assigns a resource, it is called at step 500. Two parameters, i.e., the demanded amount (REQAMT) of resources and consumption anticipation time amount (COMPT), are offered. About disk I/O, REQAMT is the number of disk I/O, and COMPT is anticipation time amount which disk I/O completes. It is confirmed whether whether an access control list compulsion means' determining the line in the execution-time physics resource table 300 about this resource, and the maximum permissible dose's 320 being specified at step 505, and the thing which actually added REQAMT to the amount used 330 exceed the maximum permissible dose 320. If it exceeds, it will display the purport which is not allowed consumption at step 527. When not exceeding the maximum permissible dose 320, an access control list compulsion means calculates the estimated consumption rate of this resource at step 510 according to / (actually amount-used 330 + REQAMT) (this time - code initiation time 395 + COMPT). At step 515, it is confirmed whether an access control list compulsion means has whether the maximum-permissible specific consumption 315 is specified and the planned specific consumption larger than the maximum-permissible specific consumption 315. If the maximum-permissible specific consumption 315 is not specified or an estimated consumption rate is not larger than it, an access control list compulsion means displays the purport allowed consumption, and is completed. It is the /(actually amount-used 300 + REQAMT) maximum-permissible specific consumption 315 about delay required to perform [for an access control list compulsion means to be step 530 if an estimated consumption rate is large, and] this actuation. - (this time -code initiation time 395 - COMPT) While it follows and calculates and only the calculated time delay displays the purport for which consumption must be delayed, a required time delay is returned.

[0025] After a resource is consumed, an access control list compulsion means is called at step 550 using the parameter which specifies resource consumption (CONSAMT). The called access control list compulsion means actually updates the amount used 330 with an activity ratio 325.

[0026] Furthermore, a different resource and permissive conditions can be assigned to a different user by the reception system about the same code. It can be performed by combining a privilege with the resource and permissive conditions which a different user was allowed by the access control list compulsion means at the code paying attention to the granted privilege, when a code is

installed. In this case, probably, the set of a resource and permissive conditions needs to memorize separately for every user. When the code is performed, a resource and permissive conditions will be forced with the user base. On the other hand, when determining a resource and permissive conditions during activation of a code, it can determine by combining a privilege with the resource and permissive conditions which a different user was allowed by the access control list compulsion means at the code paying attention to the granted privilege.

[0027] Drawing 7 is a pseudocode which shows the initialization action of integrity inspection conducted by the client system, activation, and compulsion. Note that instantiation of the DS of various kinds of tables explained until now, a list, and a flag and others is carried out in the memory (for example, volatile random access memory, a disk, or these should put together) of a client system. As mentioned above, as for an access control list compulsion means and an access control list management tool, it is desirable to carry out as a program code. It is incorporated whether these program codes are linked to the operating system of a client system. Drawing 8 shows the pseudocode of an access control list management tool. Drawing 9 shows the pseudocode of an access control list compulsion means.

[0028] Although this invention has been explained based on an example, it is thought that various kinds of modification and improvements by this contractor are possible. Therefore, the example of this invention is a mere example and he should understand what must not be interpreted restrictively. It cannot be overemphasized that the range of this invention is limited by the generic claim.

[0029] As a conclusion, the following matters are indicated about the configuration of this invention.

(1) The distribution approach of the program code which is the distribution approach of a program code and is characterized by to describe a resource and permissive conditions required for the harmless actuation in which the above-mentioned program code was inspected possible [reading] by computer in certification of the above-mentioned third person by whom reliance is done including the step which offers the certification of the third person trusted to a reception system.

(2) The distribution approach of the program code which contains the step which encapsulates the above-mentioned certification with a program code in the above (1).

(3) The distribution approach of the program code which contains the resource of the above-mentioned reception system, and the step which assigns permissive conditions in the above (1) so that the step in which the above-mentioned reception system reads the above-mentioned certification, and the permissive conditions by which it was specified during the above-mentioned certification may not be exceeded.

(4) The distribution approach of the program code containing the step which denies or affirms access and permissive conditions of the above-mentioned program code to the resource of the above-mentioned reception system in the above (2) according to the option which the user chose in relation to the above-mentioned certification.

(5) The distribution approach of the program code characterized by offering the above-mentioned resource described possible [reading] by computer, and permissive conditions in a code format in

the above (1) in the above-mentioned reception system.

(6) The distribution approach of a program code that the enciphered inspection data are contained in the above-mentioned certification in the above (1), and the above-mentioned reception system is characterized by decoding the above-mentioned inspection data and inspecting the integrity of description of the above-mentioned resource and permissive conditions.

(7) The distribution approach of the program code characterized by describing both the amount of each resource used by the above-mentioned program code into required description of a resource, and the maximum-permissible specific consumption of the resource in the above (3).

(8) The distribution approach of the program code characterized by describing the specific function of the above-mentioned reception system accessed by the above-mentioned program code into required description of permissive conditions in the above (3).

(9) The distribution approach of the program code characterized by describing the functionality of the above-mentioned program code in certification of the above-mentioned third person according to inspection by the above-mentioned third person in the above (1).

(10) The distribution approach of the program code characterized by the above-mentioned program code being the applet downloaded as a program object from a server in the above (1).

(11) The distribution approach of the program code characterized by assigning a resource which is different in the above (1) to the user from whom the above-mentioned program code differed in the above-mentioned reception system, and permissive conditions.

(12) The distribution approach of a program code that the group of the resource which a user different the account of a top is allowed in the above (11), and permissive conditions is characterized by being determined at the time of install of the above-mentioned program code.

(13) The distribution approach of a program code that the group of the resource which a user different the account of a top is allowed in the above (11), and permissive conditions is characterized by being determined at the time of activation of the above-mentioned program code.

(14) In the certification of the program code by the third person who is the distribution approach of a program code and is trusted What described a resource and permissive conditions required for the harmless actuation in which this program code was inspected possible [reading] by computer is put in. The step which encapsulates the certification with the above-mentioned program code, and is offered to a reception system in a code format, The step in which the above-mentioned reception system reads the certification of the above-mentioned code format, The account of a top The step which determines the integrity of the read certification by the above-mentioned reception system, The step which assigns the resource and permissive conditions of the above-mentioned reception system according to the option which the user chose so that the permissive conditions specified in the above-mentioned certification might not be exceeded, after the above-mentioned integrity was inspected, the step which performs the above-mentioned program code according to the above-mentioned assignment -- containing -- the above -- in required description of a resource Both the amounts and maximum-permissible specific consumption of each resource used by the above-mentioned program code are described. In description of the above-mentioned permissive

conditions The distribution approach of the program code characterized by describing the specific function of the above-mentioned reception system accessed by the above-mentioned program code.
(15) The import equipment which imports a program and data to computer system, The operating system which controls actuation of the above-mentioned computer system, The access logic which extracts what described the resource required for the harmless actuation in which the code was inspected possible [reading] by computer from the above-mentioned data, and relates it with a given program, The integrity inspection logic which generates the inspection data in which integrity is shown although it was contained in this access logic and the above-mentioned computer described possible [reading], Connect with the above-mentioned operating system, answer the above-mentioned inspection data, and consumption and specific consumption are pursued and assigned in the above-mentioned reception system about each of many resources. Computer system possessing the compulsive logic it is made not to exceed the assignment specified in the above-mentioned description, and the processor which performs a program code according to the above-mentioned assignment.
The DS memorized by random access memory is connected to the above-mentioned compulsion logic in the above (15). (16) In this DS The 1st field which pursues the consumed actual resource, and the 2nd field which pursues the specific consumption of a resource, Computer system characterized by including the 3rd field which memorizes the limit of the resource consumption pulled out from the above-mentioned description, and the 4th field which memorizes the limit of the resource specific consumption pulled out from the above-mentioned description.
(17) Computer system characterized by including a means to un-encapsulate the above-mentioned description in the above (15) from the package which contains a program code in the above-mentioned access logic, and a means to decode this description.
(18) Computer system characterized by including a means to deny or affirm code access to the resource of computer system, and permissive conditions in the above (15) according to the option which the user chose in relation to the above-mentioned description into the above-mentioned compulsion logic.

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]
[Drawing 1] It is the block diagram showing code distribution / certification system according to this invention.
[Drawing 2] In order that each part of a client system may perform the function explained in the example, it is drawing showing how it operates mutually.
[Drawing 3] It is drawing showing an access control list (ACL).
[Drawing 4] It is drawing showing the DS used by the access control list (ACL) compulsion means.
[Drawing 5] It is drawing showing how the permissive conditions of a logic resource are forced by the access control list (ACL) compulsion means.

[Drawing 6] It is drawing showing how the limit of a physical resource is forced by the access control list (ACL) compulsion means.

[Drawing 7] It is drawing having shown the initialization action of integrity inspection of a client system, activation, and compulsion by the pseudocode.

[Drawing 8] It is drawing showing the pseudocode of an access control list (ACL) management tool.

[Drawing 9] It is drawing showing the pseudocode of an access control list (ACL) compulsion means.

[Description of Notations]

10 Code Creation System

15 Certification Engine

20 Server

30 Client System

100 Code / Access Control List, and Certification

110 Inspection Means

120 Refusal Step

130 Access Control List Management Tool

140 Code

150 Access Control List

160 Authorization Flag

162 CPU and Related Control Logic

170 Activation Means

180 Access Control List Compulsion Means

190 Client Interface

194 Communication Link Controller / Network Interface

196 CD-ROM / Floppy Disk Subsystem

200 Physical Resource Table

205 Physical Resource Name

210 Resource Attribute

215 Maximum-Permissible Specific Consumption

220 Maximum Permissible Dose

250 Logic Resource Table

255 Logic Resource Name

260 Parameter List

265 Parameter Entry

270 Parameter Type

275 Parameter Value

280 NextPE (Following Parameter Entry) Field

300 Execution-Time Physics Resource Table

305 Physical Resource Name

310 Resource Attribute

315 Maximum-Permissible Specific Consumption

320 Maximum Permissible Dose

325 It is Actually Specific Consumption.

330 It is Actually the Amount Used.

350 Execution-Time Logic Resource Table

355 Logic Resource Name

360 Parameter List

365 Execution-Time Parameter Entry

370 Parameter Type

375 Parameter Value

380 NextPE (Following Parameter Entry) Field

385 Authorization Flag

395 Code Initiation Time
